EINDHOVEN UNIVERSITY OF TECHNOLOGY

Project Non-linear Optimization (2DME20)

Description and Purpose

Department of Electrical Engineering and Department of Mathematics and Computer Science

Object classification in digital images

The digital processing of images is of key importance in many engineering problems where automated diagnostics, detection, object recognition, image enhancement, encoding or classification of objects in a digital image is the main purpose. This purpose is achieved by parsing a digital image as input data through an algorithm, often in real time applications. This project will focus on the *detection* and *classification* of objects in an image. This problem amounts to finding and identifying one or multiple specific objects in an image or in a sequence of images.

Humans can recognize multiple and complex structured objects in an image without serious effort, but in the field of computer vision the unsupervised recognition and localization of objects is a difficult technical problem. An object classification algorithm takes a digital image as its input and outputs classes and locations of objects in the image. Applications of automatic object classification in images are widespread and include photography, photogrammetry, sensorics, optical character recognition, facial recognition, defect detection, security systems, car driving security, crowd and traffic control, surveillance, etc.

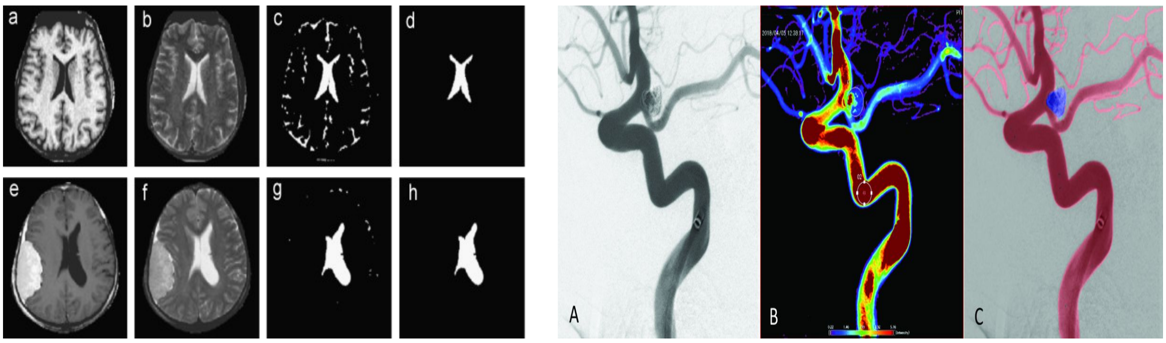

Figure 1: Feature extraction in different medical images

An image can be viewed as a two-dimensional signal in which the color or grey level intensity is a function, say f, of the spatial coordinates in the frame of the image. Specifically, if (x, y) are the spatial coordinates of a point in an image (picture, frame or digital scan), then the real number $f(x, y)$ may represent the grey level of the image at point (or pixel) (x, y). For color images $f(x, y)$ becomes a vector representing, for example, the RGB intensities at the point or pixel (x, y) in the image.

Here, we consider the classification problem to find and identify *elliptical objects* in an image. An arbitrary ellipse in a two-dimensional space can be described by the set

$$\mathcal{E} = \{ \xi \in \mathbb{R}^2 \mid (\xi - c)^\top P(\xi - c) = r^2 \} \tag{1}$$

where $c \in \mathbb{R}^2$ is the *center point* of the ellipse, $P = P^\top \succ 0$ is a positive definite matrix that represents the orientation of the principal axes of the ellipse and $r > 0$ is a constant. For example, the unit circle with center point c is obtained by setting $P = I$ and $r = 1$ in (1).

Suppose that an arbitrary image $f : \mathbb{X} \times \mathbb{Y} \to \mathbb{R}$ is defined on a rectangular domain with $\mathbb{X} = [-X/2, X/2]$ and $\mathbb{Y} = [-Y/2, Y/2]$. The image has therefore size $X \times Y$. The following steps are necessary for the automatic detection and classification of objects in f.

Step 1: Preprocessing

Usually the input image is first preprocessed to normalize contrast and brightness properties, to enhance resolution, apply cropping, transform it to a standardized color or grey level representation, and to numerically prepare the representation of the image in a useful format.

Step 2: Feature extraction

After preprocessing, the input image usually has too much redundant information for classification. The process of *feature extraction* amounts to simplify the image by retaining all relevant information for the specific classification task, while filtering or discarding all non-essential information. In particular, feature extraction techniques result in compressed and simplified digital images that are dedicated for specific object recognition tasks.

A feature extraction algorithm converts an input image to a *feature image*. This is a digital image f_{feature} that precisely represents the information that is necessary to solve the classification problem. For the classification problem that we consider here, a *feature image* is a mapping $f_{\text{feature}} : \mathbb{X} \times \mathbb{Y} \to \{0, 1\}$ defined as

$$f_{\text{feature}}(x, y) = \begin{cases} 1 & \text{if } (x, y) \in \mathcal{F} \\ 0 & \text{elsewhere} \end{cases} \tag{2}$$

where $\mathcal{F} = \{(x_i, y_i), i = 1, \ldots, N\}$ is a finite set of distinct points in $\mathbb{X} \times \mathbb{Y}$ called the *feature set*.

In this project the feature set $\mathcal{F}$ will identify the *edges* in the image. These are the points $(x, y) \in \mathbb{X} \times \mathbb{Y}$ where the image brightness $f(x, y)$ changes sharply or, more precise, has a discontinuity or a large gradient.

The automatic detection of edges in an image is one of the fundamental problems in image processing and is efficiently carried out by first low-pass filtering the image f to a smoothed image f_{smooth}. This smoothing process prevents the amplification of image noise and is usually implemented by a two-dimensional convolution

$$f_{\text{smooth}}(x, y) = (G * f)(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} G(x - \xi, y - \zeta) f(\xi, \zeta) \mathrm{d}\xi \, \mathrm{d}\zeta \tag{3}$$

where $*$ denotes convolution and $G(x, y)$ is a 2 dimensional Gaussian function $G(x, y) = g(x)g(y)$ with

$$g(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp(-\frac{x^2}{2\sigma^2})$$

for some variance parameter $\sigma > 0$. This convolution filters out isolated points and high-frequent noise from the image, but also causes blurring of the image. An *edge* is now defined by a point (x, y) for which the norm $\|\nabla f_{\text{smooth}}(x, y)\|$ of the gradient

$$\nabla f_{\text{smooth}}(x, y) = \begin{pmatrix} \frac{\partial f_{\text{smooth}}}{\partial x}(x, y) \\ \frac{\partial f_{\text{smooth}}}{\partial y}(x, y) \end{pmatrix}$$

is large while the absolute value of the second order derivative

$$\begin{aligned} \nabla^2 f_{\text{smooth}}(x, y) &= \frac{\partial^2 f_{\text{smooth}}}{\partial x^2}(x, y) + \frac{\partial^2 f_{\text{smooth}}}{\partial y^2}(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} M(x - \xi, y - \zeta) f(\xi, \zeta) \mathrm{d}\xi \, \mathrm{d}\zeta \\ &= (M * f)(x, y) \end{aligned}$$

is small. Here, the convolution kernel M is directly derived from G and equals

$$M(x, y) = \frac{x^2 + y^2 - 2\sigma^2}{\sigma^4} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right)$$

a function that is generally known as the *Mexican hat*. A plot of the function is given in Figure 2 and clearly explains its name.

This algorithm is one particular version of the *Canny edge detector* developed by John F. Canny. The *feature image* is now defined by (2) where the feature set $\mathcal{F}$ consists of the detected edges. That is,

$$\mathcal{F} := \{(x, y) \in \mathbb{X} \times \mathbb{Y} \mid \|\nabla f_{\text{smooth}}(x, y)\| > d_1 \text{ and } \|\nabla^2 f_{\text{smooth}}(x, y)\| < d_2 \} \tag{4}$$

for suitable threshold values $d_1 > 0$ and $d_2 > 0$.

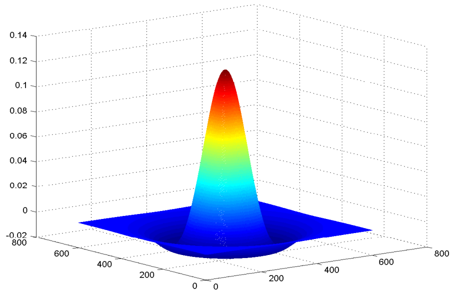

Figure 2: Convolution kernel M: the Mexican hat

Step 3: Classification of objects

The classification problem now amounts to finding one or more ellipses $\mathcal{E}$ that optimally fit the points in the feature set $\mathcal{F}$ of the feature image. This requires an optimization criterion that represents the distance between the feature set $\mathcal{F}$ and a parametrization of (one or more) ellipses $\mathcal{E}$ in the feature image f_{feature}.

In the classification, the feature image f_{feature} is therefore used as input to an optimization algorithm that determines the best fitting of ellipses in the feature image. Output of the classification algorithm are the parameters (P, c, r) that define orientation, volume and center of each fitted ellipse of the form (1).

For the classification of multiple objects in an image, the feature set $\mathcal{F}$ is often decomposed in *clusters*. That is, a *clustering procedure* is applied to decompose $\mathcal{F}$ as

$$\mathcal{F} = \mathcal{F}_1 \cup \mathcal{F}_2 \cup \cdots \cup \mathcal{F}_n$$

where $\mathcal{F}_i \cap \mathcal{F}_j = \emptyset$ for all $i \neq j$ and where every $\mathcal{F}_i$ corresponds to extracted features in a specific segment of the domain $\mathbb{X} \times \mathbb{Y}$ of the image frame. The classification problem is then distributed over all n clusters and amounts to identifying objects (ellipses in our case) in each cluster. The automated clustering amounts to finding n and partitioning data in the feature set $\mathcal{F}$ in an optimal way so as to maximize the number of classified objects in the image f.

Purpose

It is the purpose of this project to automatically detect and classify one or multiple ellipses in an arbitrary digital image of your choice. To do this, we consider different optimization criteria.